Analysis of Learning Dynamics on Heterogeneous Robot Datasets with Suboptimal Data

Haruki Abe[1,2], **Takayuki Osa**[2], **Yusuke Mukuta**[1,2], **Tatsuya Harada**[1,2]
[1]The University of Tokyo [2]RIKEN Center for Advanced Intelligence Project
Tokyo, Japan

Abstract: Robot foundation models promise versatile control across diverse embodiments. Training a single policy on heterogeneous robot data can accelerate adaptation, reduce per-platform engineering, and improve sample efficiency. However, realizing this promise is constrained by the high cost of collecting expert demonstrations at scale. We investigate a path forward by combining offline reinforcement learning (offline RL) with cross-embodiment learning to leverage datasets that mix expert and suboptimal trajectories across many morphologies, and we introduce a new locomotion benchmark that spans 16 simulated robots and multiple data-quality tiers. Our study confirms the expected benefits, namely that offline RL can make use of suboptimal data and cross-embodiment pre-training can speed adaptation to unseen robots. The central result is a failure mode. As both morphology diversity and the fraction of suboptimal trajectories grow, performance degrades for specific embodiments, particularly when similar morphologies are underrepresented in the pool. Gradient-level diagnostics trace this negative transfer to inter-robot gradient conflicts, which indicates that naïve joint training can suppress useful updates. These findings position offline RL combined with cross-embodiment learning as a promising route toward scalable robot foundation models while highlighting the need for conflict-aware optimization and embodiment-aware data curation.

Keywords: Offline Reinforcement Learning, Cross-Embodiment Learning, Locomotion

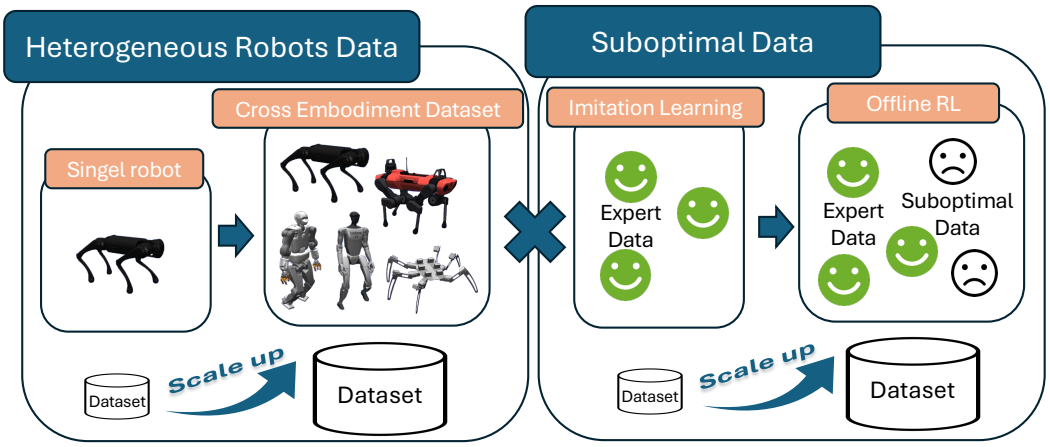


Figure 1: Offline RL + cross-embodiment learning at scale. By pooling demonstrations across heterogeneous robots, we increase both the amount and diversity of training data; adding suboptimal (non-expert) trajectories further scales this pool. Offline RL can exploit such mixed-quality data to learn, whereas imitation learning typically depends on high-quality expert demonstrations.

1 Introduction

A central question in building more general and capable robots is how to leverage learning at scale. Performance gains from scaling are now well established in NLP and vision: large language models (LLM) and vision-language models (VLM) pre-trained on web scale, diverse corpora can tackle a wide range of linguistic and visual tasks [1, 2], and generative models for images, video and music have achieved unprecedented quality [3, 4].

This momentum is increasingly influencing robotics. By scaling transformer-based architectures and training on large heterogeneous robot datasets, researchers have proposed 'robot foundation models' that address multiple tasks within a single model [5, 6, 7, 8], pointing toward general-purpose control across tasks and embodiments. However, the main constraint is the data: Compared to the massive text and image corpora that underlie today's foundation models, robot data are orders of magnitude smaller. Collecting manipulation data is time and cost intensive, requiring careful teleoperation, specialized hardware, and often manual labeling; as tasks and platforms proliferate, this burden is compounded.

A natural response to this bottleneck is cross-embodiment learning, which pre-trains a single model on demonstrations gathered from many robot platforms. Pooling heterogeneous data enables the extraction of more general control primitives, and prior work reports that multi-robot training can surpass single-robot training [6, 8]. However, most robot foundation models to date rely on imitation learning, leaving the core cost of acquiring high-quality demonstrations unresolved.

Offline reinforcement learning (offline RL) offers a complementary lever. Beyond expert demonstrations, offline RL can exploit suboptimal trajectories and improve policies despite variability in data quality. Indeed, applying offline RL to large datasets rich in suboptimal rollouts has been shown to outperform behavioral cloning [9].

Combining cross-embodiment learning with offline RL promises to substantially expand the usable pre-training pool by unifying expert and suboptimal data across embodiments. Nevertheless, this combination remains underexplored. For example, Nakamoto et al. [10] applied offline RL on two platforms without analyzing cross-embodiment effects, and Springenberg et al. [11] focused on two manipulators and toy tasks, leaving open the benefits and challenges of simultaneous pre-training over many distinct embodiments.

In this work, we present the new benchmark and analysis of pre-training that combines offline reinforcement learning (offline RL) with cross-embodiment learning, using data collected from up to 16 distinct robot platforms. Our experiments highlight two main findings. First, the combination of cross-embodiment pre-training with offline RL outperforms behavioral cloning under conditions with abundant suboptimal data, while also producing models that can rapidly adapt to unseen robots. Second, as both the proportion of suboptimal data and the number of robot types increase, performance improvements become difficult to achieve for certain robots when using standard offline RL methods. Furthermore, gradient similarity analysis reveals that inter-robot gradient conflicts are the primary driver of this negative transfer, underscoring the necessity of conflict-aware optimization when combining offline RL with cross-embodiment learning.

This paper makes the following four contributions:

1. We introduce and analyze the new benchmark that combines offline RL and cross-embodiment learning, spanning up to 16 robot platforms.
2. We demonstrate that cross-embodiment pre-training with offline RL surpasses behavioral cloning under suboptimal data conditions and accelerates adaptation to unseen robots.
3. We show that the difficulty of learning in offline RL with cross-embodiment dataset settings increases with both the proportion of suboptimal data and the number of robot types.
4. We identify inter-robot gradient conflicts as the key factor underlying performance degradation in high suboptimal data regimes, highlighting an important direction for future research.

Our experiments reveal both the benefits and the limits of leveraging offline RL in cross-embodiment settings. We see this benchmark and analysis as a concrete step toward large-scale training for robot foundation models.

2 Related Works

2.1 Offline RL

Offline reinforcement learning (RL) seeks to learn a policy that maximizes cumulative reward solely from a static dataset of interactions, without any further environment access [12]. In contrast to behavioral cloning (BC), which imitates logged behavior, offline RL can compose high-value behavior by stitching together fragments from mixed-quality datasets, often outperforming pure imitation when demonstrations include both expert and suboptimal trajectories [13, 14]. By optimizing objectives that remain anchored to the data distribution and penalize unsupported actions, these methods mitigate distribution shift and extrapolation error while still extracting policies that improve on the average quality of the dataset [13, 14]. In this work, we adopt offline RL as a pre-training paradigm for robot learning, enabling us to capitalize on broad and imperfect demonstration corpora, especially suboptimal rollouts that contain recoverable structure, and to furnish strong initial policies for subsequent foundation-model-style training in robotics.

2.2 Cross-embodiment Learning

Cross-embodiment learning trains a single network on demonstration data from multiple robot morphologies, enabling transfer of control priors across platforms. Since collecting large datasets for any single robot is costly, requiring expert teleoperation, specialized hardware, and manual labeling, pre-training on heterogeneous robot data has become a popular strategy to improve sample efficiency and generalization [5, 6, 7]. However, existing cross-embodiment foundation models rely almost exclusively on imitation learning [5, 6, 7], and there has been little work combining cross-embodiment pre-training with offline RL. While Nakamoto et al. [10] applied offline RL to data from two robot platforms, they did not analyze any cross-embodiment effects. Similarly, Springenberg et al. [11] conducted offline RL on a dataset comprising two manipulators and several toy tasks but did not investigate the benefits or challenges of learning from many distinct embodiments simultaneously. To fill this gap, we introduce the new benchmark that systematically combines offline RL with cross-embodiment learning, analyze the interactions between these paradigms, and propose methods to mitigate the challenges that arise when pooling heterogeneous and often suboptimal robot data.

3 Experimental setup

3.1 Problem Setting

We study multi-task offline RL in a heterogeneous robotics domain, where each "task" corresponds to controlling a distinct robot morphology under a common state–action interface. Concretely, let $\mathcal{T}$ denote a finite set of robot platforms (e.g., different quadrupeds, bipeds, hexapods) and let $f^{\text{morph}} : \mathcal{T} \to \mathbb{R}^{d_m}$ map each task τ to a morphology descriptor f^{morph}_τ. Each task $\tau \in \mathcal{T}$ induces an MDP

$$\mathcal{M}_\tau = \big(\mathcal{S}, \mathcal{A}, P_\tau(\cdot \,|\, s, a), r_\tau(s, a), \gamma\big), \tag{1}$$

where $\mathcal{S} \subset \mathbb{R}^{d_s}$ is the shared observation space (joint angles, velocities, etc.), $\mathcal{A} \subset \mathbb{R}^{d_a}$ is the continuous control space, $P_\tau(s' \,|\, s, a)$ is a task-specific environment dynamics, $r_\tau(s, a)$ is a dense task-specific reward and $\gamma \in (0, 1)$ is discount factor.

We assume access to a pooled offline dataset

$$\mathcal{D} \;=\; \bigcup_{\tau \in \mathcal{T}} \{(s_t, a_t, s_{t+1}, r_t, d_t)\}_{t=1}^{N_\tau}, \tag{2}$$

generated by an unknown behavior policy $\pi_\beta(a \mid s, f^{\text{morph}}(\tau))$. Our goal is to learn a single parameterized policy $\pi_\theta(a \mid s, f^{\text{morph}}(\tau))$ that maximizes the expected cumulative discounted return over

all tasks $\mathcal{R} = \sum_{\tau} \sum_{t=0}^{T} \gamma^t r(s_t, a_t)$, using only $\mathcal{D}$. Rather than providing the policy with a one-hot task index, we rely on morphology features (size, mass, link lengths, etc.), which we formalized as the descriptor $f^{\text{morph}}(\tau)$. In this way, the same policy $\pi_\theta(a \mid s, f^{\text{morph}}(\tau))$ can generalize across tasks by conditioning on these universal state features.

3.2 Environments and dataset

To facilitate cross-embodiment pre-training under an offline RL paradigm, we constructed a new locomotion dataset within the MuJoCo [15] simulation environment, following the walking tasks of Bohlinger et al. [16]. Our dataset encompasses 16 distinct robot platforms—nine quadrupeds, six bipeds, and one hexapod—each trained via Proximal Policy Optimization (PPO) [17]. During training, we record at each time step the tuple $(s_t,\, a_t,\, s_{t+1},\, r_t,\, d_t)$ to capture state, action, next state, reward and done signals.

For each robot, we curate six variants of 1 M–step datasets, divided by motion direction (forward vs. backward) and data quality:

- **Expert data**: 1 M steps collected by rolling out the fully converged PPO policy.
- **Expert Replay data**: all interaction steps from training start until expert-level performance ($\sim 500\,\text{M}$ steps in total), uniformly subsampled to $1\,\text{M}$ steps to bound dataset size.
- **70% Suboptimal Replay data**: 700 k steps drawn from the early (suboptimal) phase of PPO training, mixed with 300 k steps from the late (expert-like) phase, totaling 1 M steps.

Each of these Expert, Expert Replay, and 70% Suboptimal Replay is provided in both forward (advancing) and backward (retreating) variants, yielding a comprehensive benchmark for evaluating offline RL combined with cross-embodiment learning. See Appendix A for dataset construction details and Appendix B for reward distributions.

3.3 Network Architecture

In this section, we present our approach to cross-embodiment learning in an offline RL setting. The central challenge is to train a single network across robots whose state and action dimensions differ. To address this, we adopt the URMA architecture proposed by Bohlinger et al. [16], which enables multiple robots to share a single policy and state-value function. Concretely, URMA factorizes each observation into an embodiment-agnostic general part o_g and a robot-specific part. For locomotion, the robot-specific stream is further split into variable-length sets of joint- and foot-level observations, $\{o_j\}_{j \in J(\tau)}$ and $\{o_f\}_{f \in F(\tau)}$. Descriptor-conditioned attention aggregates each set into fixed-size latents, which are concatenated with o_g to form a morphology-agnostic core representation. To facilitate offline RL, we further extend URMA by introducing a state–action value function. Specifically, we encode each action with an action encoder to obtain a latent action vector, which we then concatenate with the latent representation of the URMA encoder. See Appendix C for architectural details.

Learning algorithms. Unless otherwise noted, we use Implicit Q-Learning (IQL) [13] as our offline RL objective for pre-training and evaluation, and train a Behavior Cloning (BC) baseline on the same pooled data. Both methods share the URMA encoder and policy head; IQL additionally uses the Q and V critics described above. Training schedules and hyperparameters are provided in Appendix D.

4 Results

We conducted a series of experiments to systematically evaluate the combination of cross-embodiment learning and offline reinforcement learning (offline RL) in robotic control. Our evaluation focused on three main questions:

Table 1: Comparison between BC+Cross-Embodiment (CE) and IQL+CE across datasets.

Dataset	BC + CE	IQL + CE
Expert Forward	63.31 ± 0.23	63.39 ± 0.11
Expert Backward	67.17 ± 0.03	67.10 ± 0.03
Expert–Replay Forward	49.71 ± 2.37	**54.61** ± 0.26
Expert–Replay Backward	42.87 ± 2.96	**51.86** ± 3.49
70% Suboptimal Forward	30.52 ± 6.94	**36.62** ± 2.29
70% Suboptimal Backward	**41.42** ± 1.58	38.69 ± 1.99
Mean	49.17	**52.05**

1. How does offline RL compare to imitation learning (IL) in cross-embodiment settings, particularly when training data contains suboptimal trajectories?
2. To what extent does cross-embodiment pre-training improve single-robot fine-tuning?
3. Under what conditions does cross-embodiment learning lead to positive or negative transfer across robot embodiments?

Experiments were performed on datasets from 16 distinct robots, covering locomotion tasks in both forward and backward directions. The following sections show results for each research question in turn.

4.1 Comparison of Imitation Learning and Offline RL

Here, we compare imitation learning, widely used in cross-embodiment training of foundational robot models, with offline RL. To date, applications of offline RL to robot foundation models have been rare, owing to the difficulty of learning from unlabeled interaction data; thus, a rigorous evaluation is necessary. Table 1 presents the performance for behavioral cloning (BC) and an implicit Q-learning (IQL) [13], commonly used offline RL method. On datasets with relatively uniform behavioral quality (for example, Expert Forward and Expert Backward), BC and offline RL achieve comparable results. In contrast, on datasets containing predominantly suboptimal trajectories, specifically Expert-Replay Data and 70% Suboptimal Replay Data Forward, offline RL methods surpass BC. This finding mirrors the results reported on benchmarks like D4RL [18] and confirms that offline RL remains robust even when datasets include significant suboptimal data in a cross-embodiment context.

4.2 Effects of Cross-Embodiment pre-training

We now evaluate how cross-embodiment pre-training impacts the performance of single-task fine-tuning. Figure 2 shows the learning curves for a "leave-one-out" experiment: we pre-train via offline RL on an Expert Forward dataset excluding one robot, then fine-tune that robot with pre-trained networks, comparing it to a model trained without cross-embodiment pre-training. The pre-trained model converges markedly faster. The results show that, for the quadrupedal robot Badger as well as for the bipedal robots Unitree G1 and Cassie, the network pre-trained via a cross-embodiment dataset is able to learn an effective policy more rapidly. These results demonstrate that cross-embodiment learning serves as a highly effective pre-training strategy in offline RL.

4.3 Positive and Negative Transfer in Cross-Embodiment Learning with Suboptimal Data

In this section, we first compare the final performance of models trained on each robot in isolation with models trained via cross-embodiment learning. Table 2 shows the final rewards achieved by IQL when trained on a cross-embodiment dataset for the Expert Forward and 70% Suboptimal Replay Forward conditions, along with the rewards obtained by models trained separately on each robot. In the Expert dataset, the cross-embodiment models learn just as effectively as the single-robot models.

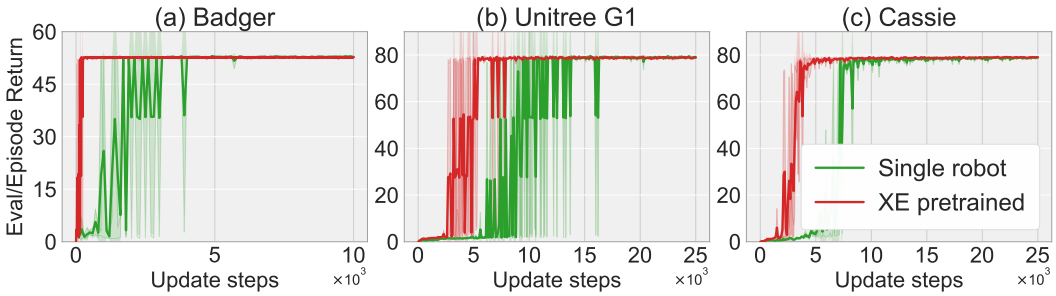


Figure 2: Comparison of learning curves between cross-embodiment pre-trained networks and networks trained without cross-embodiment pre-training for Badger, Unitree G1, and Cassie.

Table 2: Expert vs. 70% Suboptimal IQL Performance across Robots

Robot	Expert Single	Expert IQL + CE	70% Single IQL	70% IQL + CE
unitree a1	53.76	53.70	14.49	22.24
unitree go1	54.02	54.04	14.50	45.18
unitree go2	53.50	53.52	13.50	52.34
anymal b	49.82	49.75	48.12	47.77
anymal c	44.40	44.46	43.76	42.01
barkour v0	46.49	46.65	46.66	46.59
barkour vb	53.30	53.53	55.06	54.99
badger	52.96	52.90	15.97	45.33
bittle	37.05	37.04	45.75	44.95
unitree h1	54.06	53.45	54.35	0.81
unitree g1	79.02	78.76	78.48	4.08
talos	108.42	108.40	75.34	9.74
robotis op3	88.98	88.31	102.79	98.16
nao v5	83.81	83.78	61.64	69.71
cassie	79.06	79.14	1.22	1.46
hexapod	76.77	76.83	0.83	0.49
mean	63.46	63.39	42.03	36.62

Next, we analyze training on the 70% Suboptimal Replay Forward dataset, which contains a large proportion of suboptimal data. Although the average performance of cross-embodiment models falls below that of the isolated models, certain quadrupedal robots, Unitree A1, Unitree Go1, Unitree Go2, and Badger, show substantial performance gains. This suggests that positive transfer occurs among the quadrupeds, likely because they contribute the largest share of data to the dataset.

By contrast, bipedal robots such as Unitree H1 and Unitree G1, which have relatively little similar-embodiment data in the dataset, suffer pronounced performance degradation under cross-embodiment learning compared to their isolated counterparts. Because this negative transfer does not appear when using suboptimal data without cross-embodiment or when training on the Expert dataset (which contains fewer suboptimal trajectories), we conclude that negative transfer emerges only when a dataset combines large amounts of suboptimal data with the cross-embodiment training regime. In the next chapter, we further investigate the causes of this newly observed phenomenon.

Analysis: Gradient Conflicts as a Cause of Negative Transfer. We hypothesize that when suboptimal data dominate the training set, simultaneous cross-embodiment learning induces negative transfer through policy gradients conflicting gradients across robots. In particular, bipedal robots (e.g., Unitree H1 and Unitree G1) are most affected, since the dataset contains relatively little data from similarly embodied robots; these gradient conflicts can effectively cancel useful updates and stall learning.

To quantify this effect, we analyze the actor gradients arising from an AWAC/IQL-style update.[13, 19] For each embodiment τ, we define the actor objective

$$\mathcal{L}_\tau^\pi(\theta) = -\mathbb{E}_{(s,a)\sim\mathcal{D}_\tau}[w(s,a)\,\log\pi_\theta(a\mid s)]\,, \qquad w(s,a) \;=\; \exp\big(\beta(Q(s,a)-V(s))\big). \tag{3}$$

in which $Q(s,a)$ denotes the learned state-action value function and $V(s)$ denotes the learned state-value function.

We denote the per-embodiment actor gradient by

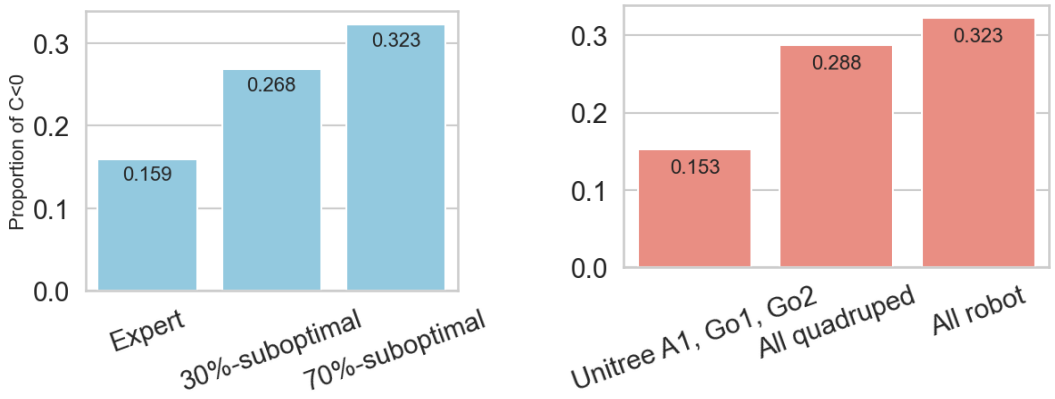
$$g_\tau \;=\; \nabla_\theta \mathcal{L}_\tau^\pi(\theta), \tag{4}$$

and measure inter-embodiment alignment via the pairwise cosine similarity

$$C[\tau_i,\tau_j] \;=\; \frac{\langle g_{\tau_i},\, g_{\tau_j}\rangle}{\|g_{\tau_i}\|\,\|g_{\tau_j}\|}. \tag{5}$$

Figure 3a reports, during training, the proportion of pairwise cosine similarities that are negative (i.e., $C[\tau_i,\tau_j] < 0$) for three datasets: Expert Forward, 30%-suboptimal-replay Forward, and 70% Suboptimal Replay Forward. As the proportion of suboptimal data increases, the share of negative cosines increases, indicating more frequent gradient conflicts between robots. This is expected because suboptimal trajectories increase the approximation and bootstrapping errors and variances in the learned critics Q and V. These errors alter the importance weights $w(s,a) = \exp\big(\beta(Q(s,a)-V(s))\big)$, thus misaligning the update directions between robots and increasing the incidence of conflicts. Furthermore, for robots whose performance changed substantially (absolute reward difference > 10) between cross-embodiment and single robot training in the 70% Suboptimal Replay dataset, we plotted the reward difference against their average gradient cosine similarity with all other robots. A strong positive correlation ($r = 0.815$) emerged: robots that exhibit positive transfer have more aligned gradients, whereas those with large negative transfer exhibit greater gradient conflict. These findings confirm that gradient conflicts underlie the negative transfer observed when applying cross-embodiment learning to datasets rich in suboptimal data.

Next, we examine how gradient conflicts change as the number and diversity of robots increase (Figure 3b). Using the 70% Suboptimal Forward dataset, we progressively expanded the set of included robots: a relatively similar group (Unitree A1, Go1, Go2), then all nine quadrupeds, and finally all 16 robots. As we include more diverse robots, each robot exhibits a higher fraction of negative pairwise cosine similarities with the others ($C < 0$). In other words, greater embodiment diversity leads to more negative cosine similarities and more frequent gradient conflicts, making negative transfer more likely. Further distributional analysis is provided in the AppendixE.



(a) Suboptimal data vs. fraction of $C < 0$.

(b) Embodiment diversity vs. fraction of $C < 0$.

Figure 3: **Fraction of negative pairwise gradient cosine similarities.** Higher values indicate stronger gradient conflicts and greater negative-transfer risk.

5 Conclusion

In this work, we presented the new benchmark and analysis of combining offline reinforcement learning with cross-embodiment learning across up to 16 distinct robot platforms and varying proportions of suboptimal data. Our experiments revealed three main findings. First, offline RL, exemplified by IQL, consistently outperforms behavioral cloning in settings where the dataset contains substantial suboptimal trajectories, confirming its robustness in cross-embodiment contexts. Second, cross-embodiment pre-training accelerates subsequent fine-tuning on unseen robots, demonstrating its effectiveness as a strategy for leveraging heterogeneous datasets to improve sample efficiency. Third, we observed that when both the number of robot types and the proportion of suboptimal data are high, performance on certain embodiments, especially those with little representation from similar morphologies, can degrade due to negative transfer.

Through gradient similarity analysis, we identified inter-robot gradient conflicts as a key mechanism underlying this degradation, establishing a quantitative link between gradient alignment and transfer outcomes. These insights highlight that while the combination of offline RL and cross-embodiment learning is a promising route toward scalable robot foundation models, mitigating gradient conflicts, particularly in the presence of abundant suboptimal data, remains an open challenge. As a priority for future work, we should develop methods to mitigate inter-robot gradient conflicts; specifically, we propose investigating optimization algorithms that are robust to such conflicts, embodiment-aware data-sampling strategies, and multi-task learning approaches that explicitly model and resolve inter-task competition, with the goal of further unlocking the potential of this paradigm.

6 Limitations

While our study demonstrates the benefits of combining offline RL with cross-embodiment learning, several limitations remain. First, all experiments were conducted in simulation on locomotion tasks; we did not evaluate on real-world robot data or deploy learned policies on physical hardware, where sensing noise, actuation latency, and safety constraints can materially affect performance and data quality. Second, our benchmark emphasizes locomotion and may not faithfully capture the characteristics of manipulation, which often involves contact-rich dynamics, object-centric observations, and different data distributions; thus, the extent to which our conclusions transfer to manipulation remains unclear. Third, although we identify inter-robot gradient conflicts as a driver of negative transfer, we leave the design and evaluation of mitigation strategies to future work. Addressing these limitations will be essential to scaling robot foundation models beyond controlled simulation settings.

Acknowledgments

This work was partially supported by JST Moonshot R&D Grant Number JPMJPS2011, CREST Grant Number JPMJCR2015, the Basic Research Grant (Super AI) of the Institute for AI and Beyond, University of Tokyo, and JSPS KAKENHI Grant Number JP25K03176.

References

[1] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[4] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[5] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[6] O. X.-E. Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[7] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[8] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[9] Y. Chebotar, Q. Vuong, A. Irpan, K. Hausman, F. Xia, Y. Lu, A. Kumar, T. Yu, A. Herzog, K. Pertsch, K. Gopalakrishnan, J. Ibarz, O. Nachum, S. Sontakke, G. Salazar, H. T. Tran, J. Peralta, C. Tan, D. Manjunath, J. Singht, B. Zitkovich, T. Jackson, K. Rao, C. Finn, and S. Levine. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *7th Annual Conference on Robot Learning*, 2023.

[10] M. Nakamoto, O. Mees, A. Kumar, and S. Levine. Steering your generalists: Improving robotic foundation models via value guidance. *Conference on Robot Learning (CoRL)*, 2024.

[11] J. T. Springenberg, A. Abdolmaleki, J. Zhang, O. Groth, M. Bloesch, T. Lampe, P. Brakel, S. Bechtle, S. Kapturowski, R. Hafner, et al. Offline actor-critic reinforcement learning scales to large models. *In International Conference on Machine Learning (ICML)*, 2024.

[12] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[13] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[14] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

[15] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[16] N. Bohlinger, G. Czechmanowski, M. Krupka, P. Kicki, K. Walas, J. Peters, and D. Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. *Conference on Robot Learning*, 2024.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[18] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[19] A. Nair, A. Gupta, M. Dalal, and S. Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

A Dataset Construction Details

This appendix describes how each dataset used in our experiments was constructed.

A.1 Expert Dataset

The *Expert* dataset was collected using an expert model that can walk almost perfectly according to a given command. For each robot, we independently trained a PPO policy to convergence and used it to generate data. Starting from the environment's default initial state, we sampled actions from the Gaussian distribution predicted by the trained policy to log trajectories. For the *Forward* dataset, we issued a command to walk forward at $1\,\mathrm{m/s}$; for the *Backward* dataset, we issued a command to walk backward at $1\,\mathrm{m/s}$.

A.2 Expert Replay Dataset

The *Expert Replay* dataset contains interaction data collected from the beginning of training up to the point at which the model can walk according to the given command (e.g., move forward at $1\,\mathrm{m/s}$ for the *Forward* condition). Storing all PPO interaction data is impractical due to PPO's low sample efficiency, which would result in extremely large data volume and step counts. We therefore constructed a uniformly thinned $1\,\mathrm{M}$-step dataset via the following procedure:

(1) Environment selection and full logging. Among the 48 parallel environments used for training, we selected one and fully recorded all rollouts (approximately $10\,\mathrm{M}$ steps) in that environment, thereby capturing trajectories from the initial exploration phase through to near convergence.

(2) Extract data up to just before convergence. From the saved logs, we reconstructed episode boundaries and computed each episode's return and length. We then applied a moving average to episode returns and identified the first point at which performance reached 90% of the final performance. Data up to just before this point were retained as the candidate set, while the subsequent steps, during which performance increases only slowly toward full convergence, were omitted.

(3) Uniform thinning to $1\,\mathrm{M}$ steps. If the total number of steps in the candidate set exceeded $1\,\mathrm{M}$, we down-sampled by discarding episodes at equal intervals with respect to cumulative steps. This preserved the overall distribution while reducing the dataset to approximately $1\,\mathrm{M}$ steps.

Using this shared procedure, we created the replay datasets for all robots. For the *Forward* datasets of Unitree A1, Go1, and Go2, the default PPO hyperparameters led to local optima. To encourage exploration and avoid such local optima, we trained these policies with an increased policy entropy coefficient, $\texttt{entropy\_coef} = 0.1$, and used the resulting data.

A.3 $X\%$ Suboptimal Dataset

The $X\%$ Suboptimal dataset is constructed so that $X\%$ of the data are suboptimal. In particular, the 70% *Suboptimal* dataset used in our experiments contains a relatively large proportion of suboptimal trajectories. Whereas the Replay dataset samples evenly from early to late training phases, the $X\%$ Suboptimal dataset is formed by sampling $X\%$ from the early training phase and $100-X\%$ from the late training phase.

B Dataset Details

Figure 4 overlays histograms of the total reward per episode for the Forward datasets, comparing the three data quality levels used throughout the paper: Expert Forward, Expert Replay Forward, and 70% Suboptimal Forward.

Overall, three consistent patterns emerge: (i) **Expert** datasets are sharply concentrated at higher returns, indicating that most episodes achieve near-target performance; (ii) **Expert Replay** exhibits a broad spread that reflects a mixture of early failures and late competent behavior accumulated during training; (iii) **70% Suboptimal** shifts mass toward lower returns, with many episodes clustered near the low-reward region.Notably, while the **70% Suboptimal** dataset contains only a small fraction of high-return episodes, those episodes tend to be considerably longer; when weighting by time steps, they account for approximately 30% of all steps. We observe qualitatively similar distributions for the Backward datasets.

These distributional differences clarify why offline RL tends to outperform pure imitation when suboptimal data are abundant: objectives that reweight actions by estimated advantages can discount low-quality behaviors while still leveraging recoverable structure present in Replay and Suboptimal corpora.

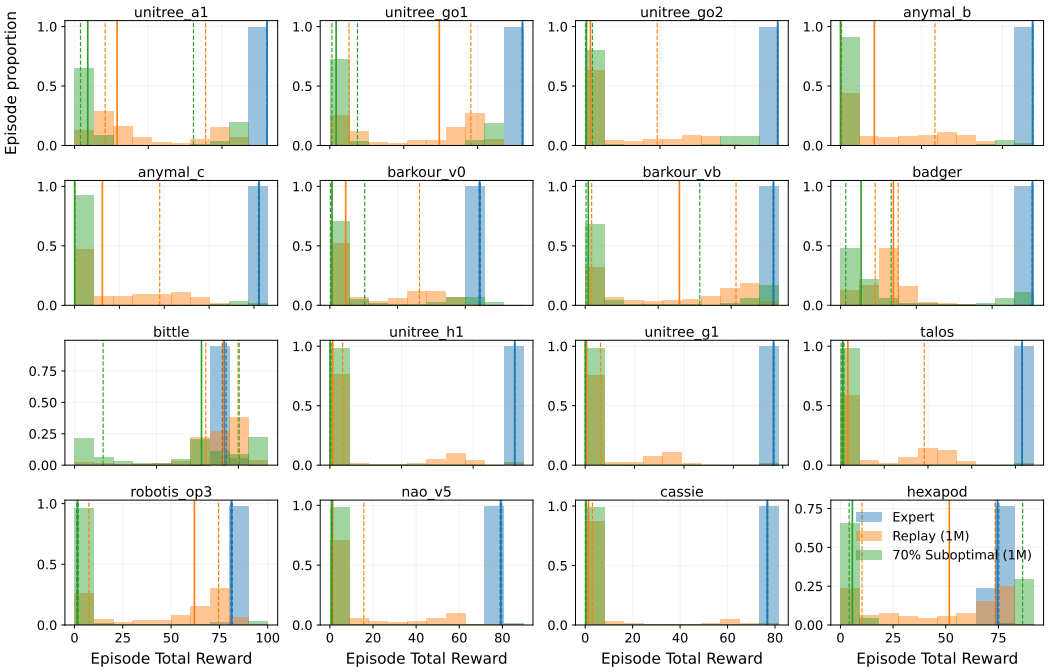


Figure 4: Overlaid histograms of per-episode total reward (x-axis) vs. episode proportion (y-axis) for Forward datasets across all robots. Each panel corresponds to a robot; colors denote Expert, Expert Replay, and 70% Suboptimal. Expert concentrates at high returns, Replay spans a wide range, and 70% Suboptimal places substantial mass on low returns.Vertical solid line: median; vertical dashed lines: 25th and 75th percentiles.

C Architecture Details

Encoder (URMA). We split each observation into general o_g and robot-specific streams. For locomotion, the latter is subdivided into joint- and foot-level sets $\{o_j\}_{j\in J(\tau)}$, $\{o_f\}_{f\in F(\tau)}$ with per-item descriptors d_j, d_f. Descriptors and observations are encoded by MLPs $f_\phi : \mathbb{R}^{\cdot} \to \mathbb{R}^{L_d}$ and $f_\psi : \mathbb{R}^{\cdot} \to \mathbb{R}^{L_d}$. URMA uses *descriptor-conditioned attention* that gates each latent *dimension*:

$$\bar{z}_{\text{joints}} = \sum_{j\in J} z_j, \qquad z_j = \frac{\exp\left(\frac{f_\phi(d_j)}{\tau+\epsilon}\right)}{\sum_{L_d} \exp\left(\frac{f_\phi(d_j)}{\tau+\epsilon}\right)} f_\psi(o_j). \tag{6}$$

In the same way, we obtain $\bar{z}_{\text{feet}}$ from $\{o_f, d_f\}$, and finally form a single latent vector by concatenation $\bar{z} = \text{concat}\big[o_g, \bar{z}_{\text{joints}}, \bar{z}_{\text{feet}}\big]$.

Actor Network. A core MLP h_θ maps the encoder output to an *action latent*:

$$\bar{z}_{\text{action}} = h_\theta(\bar{z}). \tag{7}$$

Per joint, the action head decodes Gaussian parameters from the tuple (encoded descriptor, joint latent, action latent) and samples an action:

$$a^j \sim \mathcal{N}\big(\mu_\nu(d_j^a,\ \bar{z}_{\text{action}},\ z_j),\ \sigma_\nu(d_j^a)\big), \qquad d_j^a = g_\omega(d_j).$$

Here μ_ν and σ_ν are MLPs.

State value network. The state value network use the encoder and predicts a state value from the latent:

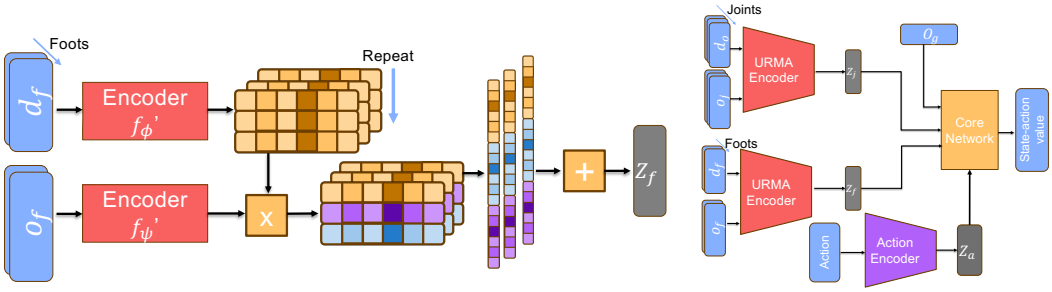
$$V_\xi(\bar{z}) \ = \ v_\xi(\bar{z}), \tag{8}$$

implemented as an MLP (e.g., $[512, 256, 128]$).

State–action value network For offline RL, we add an action encoder $f_a : \mathbb{R}^{d_a} \to \mathbb{R}^{L_a}$ and form a joint latent for Q:

$$z_a = f_a(a), \qquad Q_\eta^{(k)}(\bar{z}, a) \ = \ q_\eta^{(k)}\big(\text{concat}[\bar{z},\ z_a]\big), \quad k \in \{1, 2\}. \tag{9}$$

To accommodate heterogeneous action sizes, we feed f_a a zero-padded action vector $\tilde{a} \in \mathbb{R}^{d_{\max}}$ (padded to the maximum action length across robots).



(a) URMA encoder. Descriptor-conditioned attention aggregates joint/foot latents into a fixed-size embedding.

(b) State–action value network: The action latent vector is concatenated with other latent vectors.

Figure 5: Model overview: (a) URMA encoder; (b) State-action value network.

D Hyperparameters and Training Details

This appendix summarizes the hyperparameters and training procedures used for reproducibility. Unless otherwise noted, values are shared across all robots.

D.1 IQL and BC Hyperparameters for Training

Hyperparameters for IQL and BC are listed below.

Table 3: IQL and BC hyperparameters (common)

Parameter	IQL	BC
Discount factor γ	0.99	–
Value expectile τ_{exp}	0.7	–
Policy temperature (AWAC) β	3.0	–
Target network EMA τ_{target}	0.005	–
Batch size per robot	256	256
Learning rate	3e-4	3e-4
Offline updates	1e5	1e5
Max grad norm	0.5	0.5

D.2 PPO Hyperparameters for Dataset Generation

Representative PPO hyperparameters used for dataset collection (consistent with Appendix A).

Table 4: PPO hyperparameters (dataset collection)

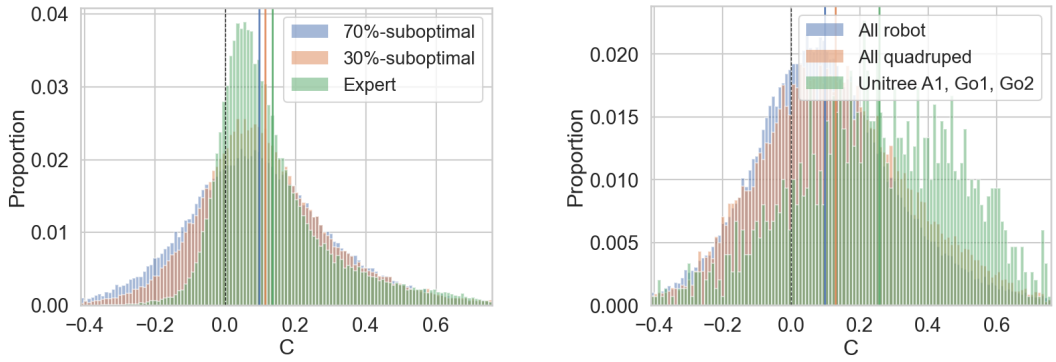
Parameter	Value
Total batch size / update	522240 (48 envs $\times$ 10880 steps)
Minibatch size	32640
SGD epochs per update	10
Learning rate (init $\rightarrow$ final)	0.0004 $\rightarrow$ 0.0 (linear anneal over 100M steps)
Entropy coefficient	0.0 or 0.1
Discount factor γ	0.99
GAE λ	0.9
Clip range (PPO ϵ)	0.1
Max gradient norm	5.0
Parallel environments	48

E Detailed analysis of gradient conflicts

We further analyze gradient conflicts by collecting the pairwise cosine similarities $C[\tau_i, \tau_j]$ for all robot pairs and training steps, and aggregating them into histograms (see Figure 6). Two consistent trends emerge:

(i) Effect of suboptimal data. As the proportion of suboptimal trajectories increases, the fraction of pairs with $C < 0$ grows. Moreover, within the negative region ($C < 0$), the share of values close to -1 increases with the proportion of suboptimal data, indicating a stronger misalignment per update and a higher probability of negative transfer. Consistently, the mean cosine $\bar{C}$ decreases as the suboptimal fraction increases. Overall, these results show that increasing suboptimal data makes gradient conflicts both more frequent and more severe.

(ii) Effect of embodiment diversity. As we include more and more diverse embodiments, the fraction of pairs with $C < 0$ increases, while the share of strongly aligned pairs (large positive C) diminishes. The mean cosine $\bar{C}$ also gradually declines as the embodiment diversity increases. Together, these effects indicate that greater embodiment diversity amplifies gradient conflicts, thereby increasing negative transfer and making positive transfer less likely.



(a) Varying suboptimal-data proportion.

(b) Varying embodiment diversity.

Figure 6: **Histograms of pairwise cosine similarities** $C[\tau_i, \tau_j]$ aggregated over training. In both settings—(a) higher suboptimal-data ratios and (b) greater embodiment diversity—the fraction with $C < 0$ increases; within $C < 0$, mass concentrates at more negative values. Solid lines indicate the mean $\bar{C}$.